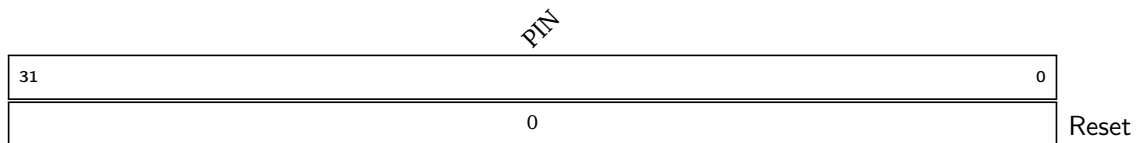


## C I/O registers

This is a list of currently supported I/O addresses. The default start address is 0xF0000000. The offset is given in bytes. Note that the I/O can only be accessed on 4-byte boundaries and on word size accesses.

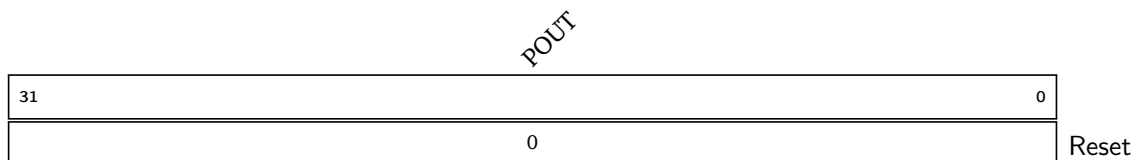
### C.1 GPIOA – General Purpose I/O



**Register C.1:** PORT A INPUT REGISTER GPIOA\_PIN (0x000)

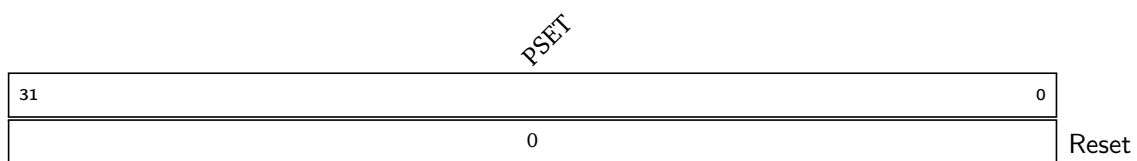
**PIN**      Port pin input values.

**Note:** This I/O register can only be read. Writes are ignored.



**Register C.2:** PORT A OUTPUT REGISTER GPIOA\_POUT (0x004)

**POUT**      Port pin output values. Write: The data is written to the output pins. Read: The last entered data is read back.



**Register C.3:** PORT A OUTPUT SET REGISTER GPIOA\_PSET (0x008)

**PSET**      Set atomically bits in the GPIOA output register. If a bit is 1, the corresponding bit in GPIOA\_POUT register is set. If a bit is 0, there is no change to the corresponding GPIOA\_POUT bit.

PCLR	
31	0
0	
Reset	

**Register C.4:** PORT A OUTPUT CLEAR REGISTER GPIOA\_PCLK (0x00c)

**PCLR** Clear atomically bits in the GPIOA output register. If a bit is 1, the corresponding bit in GPIOA\_POUT register is cleared. If a bit is 0, there is no change to the corresponding GPIOA\_POUT bit.

Reserved																PINNR				EDGE		Reserved	
31								8		7		3		2		1		0					
0										0				0		0		Reset					

**Register C.5:** EXTERNAL INPUT INTERRUPT CONTROL REGISTER GPIOA\_EXTIC (0x018)

**PINNR** Port pin number as input source.  
**EDGE** Edge selection: 00 = off, 01 = rising, 10 = falling, 11 = both.

Reserved																															DETECT	
31																															1	0
0																															0	Reset

**Register C.6:** EXTERNAL INPUT INTERRUPT STATUS REGISTER GPIOA\_EXTIS (0x01c)

**DETECT** Edge detected. Must be cleared to reset the pending interrupt.

## C.2 UART1 – Universal Asynchronous Receiver/Transmitter

Reserved																PARITY		SP2	BRIE	TCIE	RCIE	SIZE	EN
31									9	8	7	6	5	4	3	2	1	0					
0									0	0	1	1	1	0	0	Reset							

**Register C.7:** UART1 CONTROL REGISTER UART1\_CTRL (0x100)

**PARITY** 00: none, 10: even, 11: odd.

**SP2** 0: one stop bit, 1: two stop bits.  
**BRIE** BREAK received interrupt enable.  
**TCIE** Transmit character interrupt enable.  
**RCIE** Receive character interrupt enable.  
**SIZE** 00: 8 bits, 10: 9 bits, 11: 7 bits, excluding the parity.  
**EN** Enable UART1 (both receiver and transmitter)

Reserved						BR	TC	RC	PE	RF	FE
31					6	5	4	3	2	1	0
0						0	0	0	0	0	0

Reset

**Register C.8:** UART1 STATUS REGISTER UART1\_STAT (0x104)

**BR** BREAK condition detected. A BREAK is a stream of null bits for the duration of 1 start bit + number of data bits + 1 stop bit.  
**TC** Transmit completed. Set directly to 1 when a character was transmitted. Automatically cleared when writing new character to the data register or when writing 0 in the TC bit in UART1\_STAT.  
**RC** Receive completed. Set to 1 when a character was received. Automatically cleared when data register is read or when writing 0 in the RC bit in UART1\_STAT.  
**PE** Parity error. Set to 1 if parity is enabled and there is a parity error while receiving. Automatically cleared when data register is read or when writing 0 in the PE bit in UART1\_STAT.  
**RF** Receive failed. Set to 1 when failed receiving (invalid start bit). Automatically cleared when data register is read or when writing 0 in the RF bit in UART1\_STAT.  
**FE** Frame error. Set to 1 when a low is detected at the position of the (first) stop bit. Automatically cleared when data register is read or writing a 0 in the FE bit in UART1\_STAT.

Reserved									DATA		
30							9	8			0
0								0			

Reset

**Register C.9:** UART1 DATA REGISTER UART1\_DATA (0x108)

**DATA** The data to be transmitted or received.

**Write:** The data is written to an internal buffer and transmitted.

Size depends on the SIZE field in the UART1 Control Register.

**Register C.10: UART1 BAUD RATE REGISTER UART1\_BAUD (0x10c)**

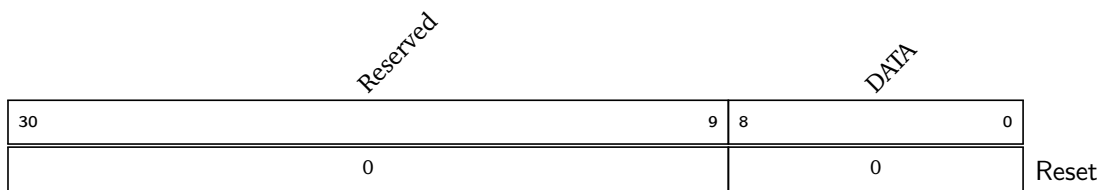
### C.3 UART2 – Universal Asynchronous Receiver/Transmitter

**Register C.11: UART2 CONTROL REGISTER UART2\_CTRL (0xb00)**

### Register C.12: UART2 STATUS REGISTER UART2\_STAT (0xb04)

4

- RC** Receive completed. Set to 1 when a character was received. Automatically cleared when data register is read or when writing 0 in the RC bit in UART2\_STAT.
- PE** Parity error. Set to 1 if parity is enabled and there is a parity error while receiving. Automatically cleared when data register is read or when writing 0 in the PE bit in UART2\_STAT.
- RF** Receive failed. Set to 1 when failed receiving (invalid start bit). Automatically cleared when data register is read or when writing 0 in the RF bit in UART2\_STAT.
- FE** Frame error. Set to 1 when a low is detected at the position of the (first) stop bit. Automatically cleared when data register is read or writing a 0 in the FE bit in UART2\_STAT.



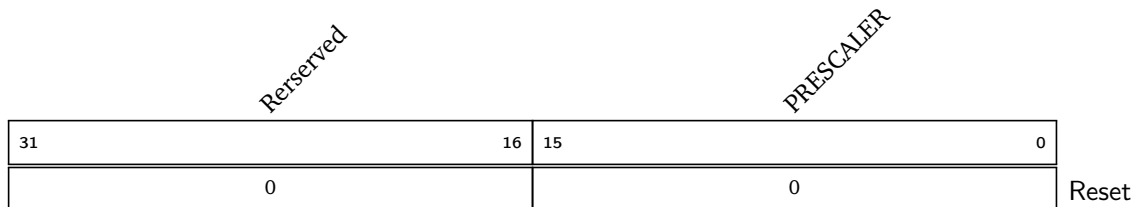
**Register C.13:** UART2 DATA REGISTER UART2\_DATA (0xb08)

**DATA** The data to be transmitted or received.

**Write:** The data is written to an internal buffer and transmitted.

**Read:** The last received data is read.

Size depends on the SIZE field in the UART2 Control Register.



**Register C.14:** UART2 BAUD RATE REGISTER UART2\_BAUD (0x10c)

$$\text{PRESCALER Baud rate} = \frac{f_{\text{system}}}{\text{prescaler} + 1}$$

## C.4 I2C1 – Inter-Integrated Circuit master-only controller

General purpose I<sup>2</sup>C peripheral, with programmable baud rate prescaler, start- and stop-bit generation, support for clock stretching, no arbitration, Standard mode (Sm) and Fast mode (Fm) only.

BAUD																Reserved				MACK HARDSTOP START STOP				Reserved				TCIE FM		Reserved		
31																16	15	12	11	10	9	8	7					4	3	2	1	0
0																0				0	0	0	0	0				0	0	0	0	Reset

**Register C.15:** I2C1 CONTROL REGISTER I2C1\_CTRL (0x200)

<b>BAUD</b>	Baud rate prescaler. Number of system clock pulses minus 1 for <b>one-half</b> bit time (Sm) or <b>one-third</b> bit time (Fm). Note: because of the 50 MHz system frequency, the lowest I <sup>2</sup> C clock frequency is 763 Hz (Sm).
<b>MACK</b>	Set to 1 to acknowledge a reception by the master. Must only be used when receiving.
<b>HARDSTOP</b>	Set to 1 to just generate a STOP condition. Useful after addressing a target that didn't respond. Cleared by hardware.
<b>START</b>	Send a START before next byte send. Cleared by hardware when transmission ends.
<b>STOP</b>	Send a STOP after next byte send or received. Cleared by hardware when transmission ends.
<b>TCIE</b>	Transmission Complete interrupt enable.
<b>FM</b>	0: Standard mode 1:1 (SCL 1/2 low, 1/2 high) 1: Fast mode 2:1 (SCL 2/3 low, 1/3 high)

Reserved																BUSY		AF		Reserved		TC		TRANS		Reserved	
31																7	6	5	4	3	2	1	0				
0																0	0	0	0	0	0	0	Reset				

**Register C.16:** I2C1 STATUS REGISTER I2C1\_STAT (0x204)

<b>BUSY</b>	Set to 1 when SDA or SCL is low, set to 0 when STOP condition is detected, independent of the I2C1 device.
<b>AF</b>	Acknowledge Fail, set when no target responded. Cleared by hardware when I2C1_DATA is accessed.
<b>TC</b>	Transmission Complete, including START or STOP, if any. Cleared by hardware when I2C1_DATA is accessed.
<b>TRANS</b>	Indicates transmitting (1) or not (0) by this controller.

Reserved																DATA											
31																8		7		0							
0																				0							

Reset

**Register C.17:** I2C1 DATA REGISTER I2C1\_DATA (0x208)

**DATA** Data to be transmitted or received.

Write: The data is written to an internal buffer and transmitted.

Read: The last received data is read.

## C.5 I2C2 – Inter-Integrated Circuit master-only controller

General purpose I<sup>2</sup>C peripheral, with programmable baud rate prescaler, start- and stop-bit generation, support for clock stretching, no arbitration, Standard mode (Sm) and Fast mode (Fm) only.

BAUD																Reserved				MACK HARDSTOP START STOP				Reserved				TCIE FM		Reserved	
31																16	15	12	11	10	9	8	7	4	3	2	1	0	Reset		
0																0		0	0	0	0	0		0	0	0					

**Register C.18:** I2C2 CONTROL REGISTER I2C2\_CTRL (0x300)

<b>BAUD</b>	Baud rate prescaler. Number of system clock pulses minus 1 for <b>one-half</b> bit time (Sm) or <b>one-third</b> bit time (Fm). Note: because of the 50 MHz system frequency, the lowest I <sup>2</sup> C clock frequency is 763 Hz (Sm).
<b>MACK</b>	Set to 1 to acknowledge a reception by the master. Must only be used when receiving.
<b>HARDSTOP</b>	Set to 1 to just generate a STOP condition. Useful after addressing a target that didn't respond. Cleared by hardware.
<b>START</b>	Send a START before next byte send. Cleared by hardware when transmission ends.
<b>STOP</b>	Send a STOP after next byte send or received. Cleared by hardware when transmission ends.
<b>TCIE</b>	Transmission Complete interrupt enable.
<b>FM</b>	0: Standard mode 1:1 (SCL 1/2 low, 1/2 high) 1: Fast mode 2:1 (SCL 2/3 low, 1/3 high)

Reserved							BUSY AF Reserved TC TRANS Reserved							
31							7	6	5	4	3	2	1	0
0								0	0	0	0	0	0	Reset

Reset

**Register C.19:** I2C2 STATUS REGISTER I2C2\_STAT (0x304)

- BUSY** Set to 1 when SDA or SCL is low, set to 0 when STOP condition is detected, independent of the I2C2 device.
- AF** Acknowledge Fail, set when no target responded. Cleared by hardware when I2C2\_DATA is accessed.
- TC** Transmission Complete, including START or STOP, if any. Cleared by hardware when I2C2\_DATA is accessed.
- TRANS** Indicates transmitting (1) or not (0) by this controller.

Reserved															DATA																																													
31															8															7															0															
0																														0																														Reset

Reset

**Register C.20:** I2C2 DATA REGISTER I2C2\_DATA (0x308)

- DATA** Data to be transmitted or received.

Write: The data is written to an internal buffer and transmitted.

Read: The last received data is read.

## C.6 SPI1 – Serial Peripheral Interface

General purpose SPI master peripheral, with prescaler, 8/16/24/32 bits data exchange and interrupt.

Reserved											PRESCALER				Reserved				SIZE		TCIE	CPOL	CPHA	Reserved		
31											11		10	8	7	6	5	4	3	2	1	0				
0											0		0		0		0		0		0		0		Reset	

Reset

**Register C.21:** SPI1 CONTROL REGISTER SPI1\_CTRL (0x400)

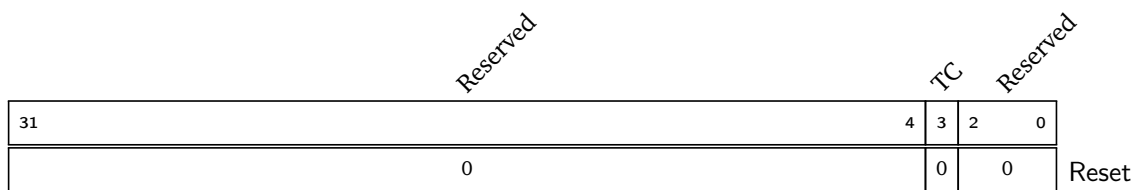
- PRESCALER** 000 /2  
001 /4



**010** /8  
**011** /16  
**100** /32  
**101** /64  
**110** /128  
**111** /256

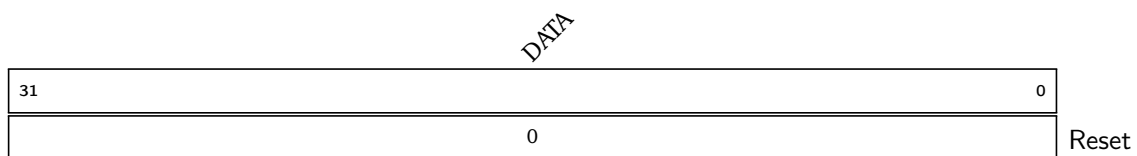
Note: because of the 50 MHz system frequency, the lowest SPI clock frequency is 195.3125 kHz.

**SIZE**      **00** 8 bits  
               **01** 16 bits  
               **10** 24 bits  
               **11** 32 bits  
  
**TCIE**      Transfer complete interrupt enable  
**CPOL**      Clock polarity  
**CPHA**      Transfer phase



**Register C.22:** SPI1 STATUS REGISTER SPI1\_STAT (0x404)

**TC**      Transfer complete



**Register C.23:** SPI1 DATA REGISTER SPI1\_DATA (0x408)

**DATA**      Data to be transmitted or received.

Write: The data is written to an internal buffer and transmitted.

Read: The last received data is read.

Data size depends on the SIZE field in the SPI1 Control Register. Data is right aligned.

## C.7 SPI2 – Serial Peripheral Interface

General purpose SPI master peripheral, with prescaler, 8/16/24/32 bits data exchange and interrupt.

Reserved											PRESCALER				Reserved		SIZE		Reserved		CPOL	CPHA	Reserved	
31											11		10	8	7	6	5	4	3	2	1	0	Reset	
0											0		0		0		0	0	0	0				

**Register C.24:** SPI2 CONTROL REGISTER SPI2\_CTRL (0x500)

<b>PRESCALER</b>	<b>000</b>	/2
	<b>001</b>	/4
	<b>010</b>	/8
	<b>011</b>	/16
	<b>100</b>	/32
	<b>101</b>	/64
	<b>110</b>	/128
	<b>111</b>	/256

Note: because of the 50 MHz system frequency, the lowest SPI clock frequency is 195.3125 kHz.

<b>SIZE</b>	<b>00</b>	8 bits
	<b>01</b>	16 bits
	<b>10</b>	24 bits
	<b>11</b>	32 bits

**CPOL** Clock polarity

**CPHA** Transfer phase

Reserved											TC		Reserved	
31											4	3	2	0
0											0	0		Reset

**Register C.25:** SPI2 STATUS REGISTER SPI2\_STAT (0x504)

**TC** Transfer complete

31	0
0	

Reset

**Register C.26:** SPI2 DATA REGISTER SPI2\_DATA (0x508)

**DATA** Data to be transmitted or received.

Write: The data is written to an internal buffer and transmitted.

Read: The last received data is read.

Data size depends on the SIZE field in the SPI2 Control Register. Data is right aligned.

## C.8 TIMER1 – a simple timer

Simple 32-bit timer peripheral for time base generation, with interrupt.

31	5	4	3	1	0
0		0	000	0	0

Reset

**Register C.27:** TIMER1 CONTROL REGISTER TIMER1\_CTRL (0x600)

**EN** Enable the timer

**TCIE** Timer compare match interrupt enable

31	5	4	3	0
0		0	0000	0

Reset

**Register C.28:** TIMER1 STATUS REGISTER TIMER1\_STAT (0x604)

**TCI** Timer compare match. Set to 1 on compare match between the timer Count register and the Compare Match register. Must be cleared by software by writing a 0.

CNTR																															
31																															0
0																															

Reset

Reset

**Register C.29:** TIMER1 COUNT REGISTER TIMER1\_CNTR (0x608)

**CNTR** This register holds the counted clock pulses on the timer. This register may be written by software.

CMPT																															
31																															0
0																															

Reset

Reset

**Register C.30:** TIMER1 COMPARE TIMER T REGISTER TIMER1\_CMPT (0x60c)

**CMPT** This register holds the value at which the counter register is compared. On CNTR compares to greater than or equal to CMPT, the counter register will be cleared and the TCI flag will be set (both in the next clock cycle).

## C.9 TIMER2 – a more elaborate timer

General purpose 16-bit timer with Output Compare, PWM generation and Input Capture capabilities, preload and interrupt (one vector).

FOCC FOCB FOCA FOCT PHAC MODEC PHAB MODEB PHAA MODEA PHAT MODET PREC PREB PREA PRET CCIE BCIE ACIE TCIE OS Reserved EN																																		
31	30	29	28	27	26	24	23	22	20	19	18	16	15	14	12	11	10	9	8	7	6	5	4	3	2	1	0							
0	0	0	0	0	0	000	0	000	0	000	0	000	0	000	0	0	0	0	0	0	0	0	0	0	0	00	0							

Reset

**Register C.31:** TIMER2 CONTROL REGISTER TIMER2\_CTRL (0x700)

**FOCC** Force Output Compare match C.  
**FOCB** Force Output Compare match B.  
**FOCA** Force Output Compare match A.  
**FOCT** Force Output Compare match T.  
**PHAC** Register C start phase.  
**MODEC** Register C mode.  
**PHAB** Register B start phase.  
**MODEB** Register B mode.  
**PHAA** Register A start phase.  
**MODEA** Register A mode.  
**PHAT** Register T start phase.

<b>MODET</b>	Register T mode.
<b>PREC</b>	Enable compare register C preload.
<b>PREB</b>	Enable compare register B preload.
<b>PREA</b>	Enable compare register A preload.
<b>PRET</b>	Enable compare register T preload.
<b>CCIE</b>	Timer compare match/input capture C interrupt enable.
<b>BCIE</b>	Timer compare match/input capture B interrupt enable.
<b>ACIE</b>	Timer compare match/input capture A interrupt enable.
<b>TCIE</b>	Timer compare match T interrupt enable.
<b>OS</b>	One-shot mode.
<b>EN</b>	Enable the timer.

If none of the FOCx bits are 1, MODET and MODEA/B/C have the following meaning:

- 000** Output off.
- 001** Toggle on compare match.
- 010** Set high on compare match.
- 011** Set low on compare match.
- 100** Edge-aligned PWM (only A/B/C, for T not allowed).
- 101** Reserved.
- 110** Input capture positive edge (only A/B/C, for T not allowed).
- 111** Input capture negative edge (only A/B/C, for T not allowed).

If at least one of the FOCx bits is 1, MODET and MODEA/B/C have the following meaning:

- 000** Not used.
- 001** Toggle output compare.
- 010** Set high output compare.
- 011** Set low output compare.
- 100** not allowed.
- 101** not allowed.
- 110** not allowed.
- 111** not allowed.

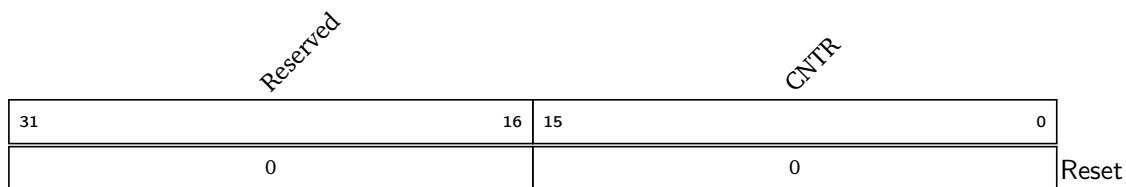
In this case, the CTRL register is not written and keeps its original setting.

Reserved								CC BC AC TC				Reserved			
31							8	7	6	5	4	3			0
0								0	0	0	0		0000	Reset	

**Register C.32:** TIMER2 STATUS REGISTER TIMER2\_STAT (0x704)

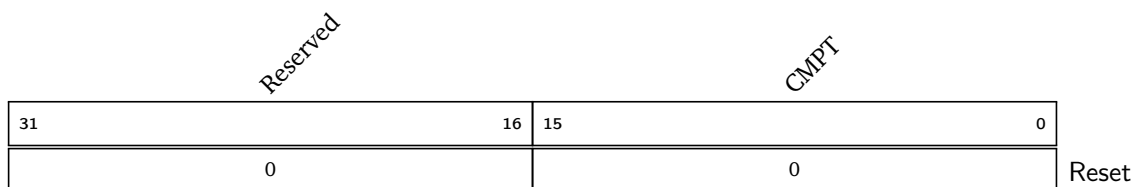
- CC** Timer compare match C. Set to 1 on compare match between the timer Count register and the Compare Match C register. Set

- on input capture on detecting selected edge. Must be cleared by software by writing a 0.
- BC** Timer compare match B. Set to 1 on compare match between the timer Count register and the Compare Match B register. Set on input capture on detecting selected edge. Must be cleared by software by writing a 0.
- AC** Timer compare match A. Set to 1 on compare match between the timer Count register and the Compare Match A register. Set on input capture on detecting selected edge. Must be cleared by software by writing a 0.
- TC** Timer compare match T. Set to 1 on compare match between the timer Count register and the Compare Match T register. Must be cleared by software by writing a 0.



**Register C.33:** TIMER2 COUNT REGISTER TIMER2\_CNTR (0x708)

- CNTR** This register holds the counted clock pulses on the timer. This register may be written by software. Rolls over when CNTR compare greater than or equal to CMPT on the next clock cycle.



**Register C.34:** TIMER2 COMPARE TIMER T REGISTER TIMER2\_CMPT (0x70c)

- CMPT** This register holds the value at which the Count register is compared. On CNTR compares to greater than or equal to CMPT, the Count register will be cleared and the TC flag will be set (both in the next clock cycle).

Reserved																PSRC																															
31																15																0															
0																0																Reset															

### Register C.35: TIMER2 PRESCALER REGISTER TIMER2\_PRSC (0x710)

**PRSC** This register holds the prescaler of the timer. This register may be written by software. Whenever the internal prescaler is equal to or greater than this register, the internal prescaler is reset. This register should only be written when the timer is stopped. Writing this register resets the internal prescaler.

Reserved		CMPA	
31	16	15	0
0		0	
		Reset	

**Register C.36: TIMER2 COMPARE TIMER A REGISTER TIMER2\_CMPA (0x714)**

**CMPA** For Output Compare: This register holds the value at which the Count register is compared. On CNTR compares to greater than or equal to CMPA, the ACI flag will be set in the next clock cycle. For Input Capture: The value of CNTR is copied to CMPA on detecting the selected edge, and the AC flag is set.

Reserved		CMPB	
31	16	15	0
0		0	
		Reset	

**Register C.37: TIMER2 COMPARE TIMER B REGISTER TIMER2\_CMPB (0x718)**

**CMPB** For Output Compare: This register holds the value at which the Count register is compared. On CNTR compares to greater than or equal to CMPB, the BCI flag will be set in the next clock cycle. For Input Capture: The value of CNTR is copied to CMPB on detecting the selected edge, and the BC flag is set.





Note: this register must be written with the WDT password to reset the watchdog. No actual value is written in this register. Reads return all zero bits.

## C.11 MSI – Machine Software Interrupt

Note: MSI has to be enabled by writing a 1 to mie.MSIE.

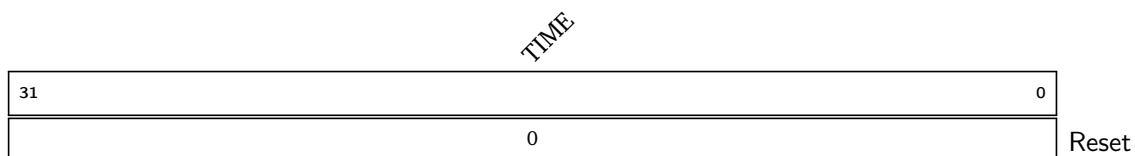


**Register C.41:** MSI TRIGGER REGISTER MSI\_TRIG (0x900)

**TRIG** Writing a 1 to this field will trigger an MSI. Writing a 0 will disarm the trigger.

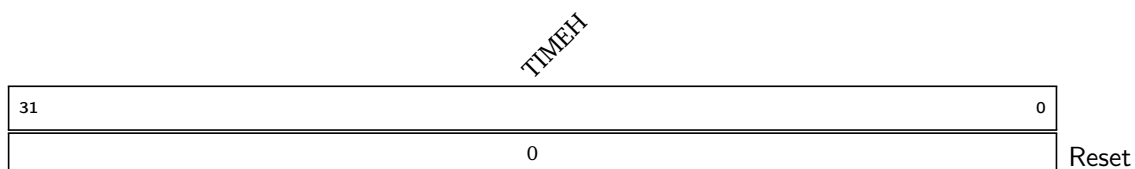
## C.12 MTIME – RISC-V system timer

Note: the external timer interrupt has to be enabled by writing a 1 to mie.MTIE. Note: the external timer will assert a pending interrupt if TIMEH:TIME (viewed as a 64-bit register) is greater than or equal to TIMECMPH:TIMECMP (viewed as a 64-bit register). To negate the pending interrupt, set TIMECMPH:TIMECMP to a higher value than TIMEH:TIME. The TIMEH:TIME registers count the number of micro seconds since last reset. As such, the system clock frequency must be a integer multiple of 1 MHz.



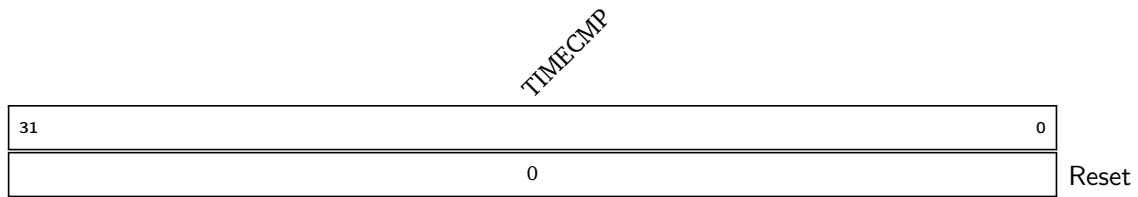
**Register C.42:** TIME EXTERNAL TIMER REGISTER TIME (0xa00)

**TIME** This register holds the low 32 bits of the external timer. Currently read-only.



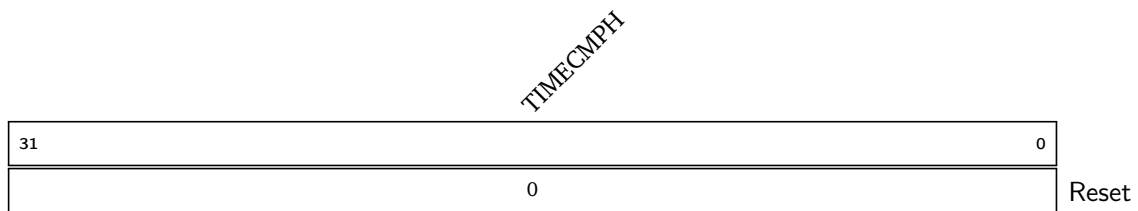
**Register C.43:** TIMEH EXTERNAL TIMER REGISTER TIME (0xa04)

**TIMEH** This register holds the upper 32 bits of the external timer. Currently read-only.



**Register C.44:** TIMECMP EXTERNAL TIMER COMPARE REGISTER TIMECMP (*0xa08*)

**TIMECMP** This register holds the low 32 bits of the external timer compare register.



**Register C.45:** TIMECMPH EXTERNAL TIMER COMPARE REGISTER TIMECMP (*0xa0c*)

**TIMECMPH** This register holds the upper 32 bits of the external timer compare register.